AMENDMENTS TO THE CLAIMS

This listing of claims will replace all prior versions, and listings, of claims in the application.

Listing of Claims:

1. (Currently Amended) A method comprising:

detecting ~~a first location-independent~~an error within ~~an area~~a cache line of a cache memory; and

~~determining whether the first error is a second consecutive error associated with the area; and~~

~~preventing further use of the area if the error is determined to be the second consecutive error associated with the area~~modifying by a cache management system to at least inhibit subsequent access to the cache line, said modifying including at least a selected one of modifying by the cache management system a value corresponding to a particular set such that the cache line is less likely to be accessed than at least one other cache line, and assigning by the cache management system a disable state to the cache line as part of a MESI state assignment.

2. (Currently Amended) The method of claim 1, wherein the ~~area~~ cache memory comprises ~~a cacheline of~~ an n-way set associative static random access memory.

3. (Currently Amended) The method of claim 2, wherein the ~~location-independent~~ error comprises an error checking and correcting (ECC) based error.

4. (Currently Amended) The method of claim 1, further comprising:

comparing a current operation address corresponding to the ~~first~~ error with a stored address corresponding to a previous error; and

identifying the ~~first~~ error as ~~the~~ a second consecutive error if the current operation address matches the stored address.

5. (Currently Amended) The method of claim 4, further comprising:

storing the current operation address corresponding to the ~~first~~ error in place of the stored address corresponding to the previous error if the current operation address does not match the stored address.

6. (Currently Amended) The method of claim 5, further comprising:

accessing the current operation address corresponding to the ~~first~~ error a second time to determine whether the second consecutive error occurs.

7-10 (Cancelled)

11. (Currently Amended) The method of claim 1, further comprising:

determining whether data stored in the ~~area~~ cache line of the cache memory has been modified as compared to data stored in a corresponding area of main memory;

facilitating recovery of the modified data stored in the area of the cache memory; and

preventing further use of the ~~area~~ cache line of the cache memory after the modified data stored in the ~~area~~ cache line of cache memory has been recovered.

12-14 (Cancelled)

15. (Currently Amended) The method of claim ~~12~~1, wherein the ~~first~~ error is detected via programmable built-in self test (PBIST) logic.

16. (Currently Amended) A processor comprising:

a cache memory;

error detection logic coupled to the cache memory, the error detection logic equipped ~~to~~ to:

detect ~~a~~ an ~~first location independent~~ error within ~~an area~~a cache line of the cache ~~memory,~~memory; and

~~determine whether the first error is a second consecutive error associated with the area; and~~

~~prevent further use of the area if the error is determined to be the second consecutive error associated with the area~~modify a cache management system to at least inhibit subsequent access to the cache line, said modifying including at least a selected one of modifying a value corresponding to a particular set such that the cache line is less likely to be accessed than at least one other cache line, and assigning a disable state to the cache line as part of a MESI state assignment.

17. (Original) The processor of claim 16, wherein the cache memory comprises an n-way set associative static random access memory (SRAM).

18. (Currently Amended) The processor of claim 16, wherein the error detection logic comprises:

error checking and correcting logic to detect an error within the cache memory;

hard error detection logic to determine whether the detected error is a hard error; and

cacheline disable logic to disable a ~~cacheline~~ cache line affected by the error if it is determined that the detected error is a hard error.

19. (Currently Amended) The processor of claim 18, wherein the hard error detection logic further comprises:

an error register to store address information indicating at least the ~~cacheline~~ cache line affected by the error;

comparison logic to compare stored address information with a current operating address; and

state logic to determine whether the error is a second consecutive error based upon output from the comparison logic.

20. (Currently Amended) The processor of claim 19, wherein the cacheline disable logic further comprises:

a decoder to generate a way-disable vector prevent further use of the ~~area~~ cache line; and

way-select logic to select the ~~cacheline~~ cache line to be disabled based upon the way-disable vector;

wherein the way-disable vector operates to modify a cache management system including at least one of a least recently used (LRU) algorithm and a MESI protocol.

21-23 (Cancelled)

24. (Currently Amended) The processor of claim ~~21~~16, further comprising programmable built-in self test (PBIST) logic coupled to the error detection logic to facilitate detection of the ~~first location independent~~ error during a startup routine of the processor.

25. (Currently Amended) A system comprising:

a dynamic random access memory; and

an integrated circuit coupled to the dynamic random access memory, the integrated circuit including a cache memory and error detection logic, wherein the error detection logic is equipped to

detect ~~a~~ an ~~first location independent~~ error within ~~an a area~~ cache line of the cache memory, and

~~determine whether the first error is a second consecutive error associated with the area, and~~

~~prevent further use of the area if the error is determined to be the second consecutive error associated with the area~~modify a cache management system to at least inhibit subsequent access to the cache line, said modifying including at least a selected one of modifying a value corresponding to a particular set such that the cache line is less likely to be accessed than at least one other cache line, and assigning a disable state to the cache line as part of a MESI state assignment.

26. (Original) The system of claim 25, wherein the integrated circuit further includes a central processing unit and at least one input/output module coupled to the central processor unit.

27. (Original) The system of claim 25, wherein the integrated circuit is a microprocessor.

28-30 (Cancelled)